Computational Control Workstation: Users' Perspectives

Carlos M. Roithmayr* and Timothy M. Straube†
NASA Johnson Space Center, Houston, Texas, 77058

Jeffrey S. Tave‡
Lockheed Engineering and Sciences Company, Houston, Texas, 77258

Abstract

A Workstation has been designed and constructed for rapidly simulating motions of rigid and elastic multibody systems. We examine the Workstation from the point of view of analysts who use the machine in an industrial setting. Two aspects of the device distinguish it from other simulation programs. First, one uses a series of windows and menus on a computer terminal, together with a keyboard and mouse, to provide a mathematical and geometrical description of the system under consideration. The second hallmark is a facility for animating simulation results. An assessment of the amount of effort required to numerically describe a system to the Workstation is made by comparing the process to that used with other multibody software. The apparatus for displaying results as a motion picture is critiqued as well. In an effort to establish confidence in the algorithms that derive, encode, and solve equations of motion, simulation results from the Workstation are compared to answers obtained with other multibody programs. Our study includes measurements of computational speed.

Introduction

A companion paper, Ref. [1], describes in detail a Workstation consisting of hardware and software designed specifically for performing numerical simulations of motions of rigid and elastic multibody systems. Through a series of windows and menus on the Workstation's console, an analyst describes a system's topography and mass distribution, provides information associated with elastic behavior of the system's bodies, and creates geometric representations of each body for animating simulation results. Dynamical equations of motion for the system of interest are derived via symbolic manipulation and an order N algorithm based on Kane's method [2], tailored to take advantage of four parallel processors, and encoded into FORTRAN subroutines. Simulation results can be displayed as numerical values, plots, or a three-dimensional animation.

We are in the process of becoming familiar with the Workstation, and suggesting ways to make it easier to use. Evaluations are presented here that are subjective and, where possible, objective in nature, based on our experiences thus far.

The manner in which multibody systems are described is addressed first. Second, we take up the means of presenting simulation results— in the form of curves, or as a motion picture.

*Aerospace Engineer, Vehicle Dynamics Section.

†Co-op Student (University of Colorado at Boulder), Vehicle Dynamics Section.

‡Engineer; Guidance, Control, and Aerosciences Dept.

Next, comparisons of results with those from other simulation programs are discussed, followed by comparisons of computational speed. Finally, a summary of suggestions for some of the more significant improvements to the Workstation is presented.

Describing a Multibody System

Before a simulation can be performed with the Workstation, one must furnish numerical information about the topography of the system of interest, the mass distribution of each body, and elastic behavior of deformable bodies. If the system's motion is to be animated, geometric representations of each body must be supplied as well.

Data-entry windows appearing on the Workstation console are the principal avenues for supplying mathematical descriptions of a system. Familiarity with a text editor is unnecessary when using the windows, which contain graphical "buttons", and well defined, labeled fields into which numerical values are typed, as shown in Fig. 1. The colors black and grey are used to distinguish relevant information from the irrelevant. For example, if a body is regarded as rigid, "Model Reduction" and all of the labels and fields related to elastic behavior become grey. Messages are produced instantly to give notification of a user's mistakes. All of the information supplied through the windows is stored in a data-entry file. The windows are intended to free analysts from having to know the format of the files; however, the goal is not altogether achieved. Moreover, the windows present a few disadvantages.

Numerical information needed to describe a system often lies scattered over several computers. Data can be "cut" from a file residing on another machine and "pasted" into a data-entry file on the Workstation when the contents of both files are displayed in separate portions of the console; however, this requires a knowledge of data-entry files' format. This kind of information exchange is not possible when data-entry windows are used— an inconvenient state of affairs.

It is easier to check a system description for errors by inspecting a single data-entry file, rather than many data-entry windows. In many cases, correcting slight errors or making minor changes is accomplished more quickly by editing a file than by using the windows. Additional motivation for working directly with data-entry files arises because the Workstation can serve (in addition to an analyst on the console) one or more remote users through a computer network, but the data-entry windows can not be displayed on remote terminals.

The data-entry windows seem most useful when a system is to be described for the first time, or when major changes are to be made. Although the files in which descriptions are stored contain many comments indicating the nature of particular numerical values, it is essential that the user's manual contain a complete explanation of valid options, data types, and proper placement.

A system composed of several rigid bodies fastened together is simpler to deal with than a system that includes deformable bodies. Therefore, the method for describing rigid bodies is reviewed first, followed by a discussion of the process by which an analyst creates

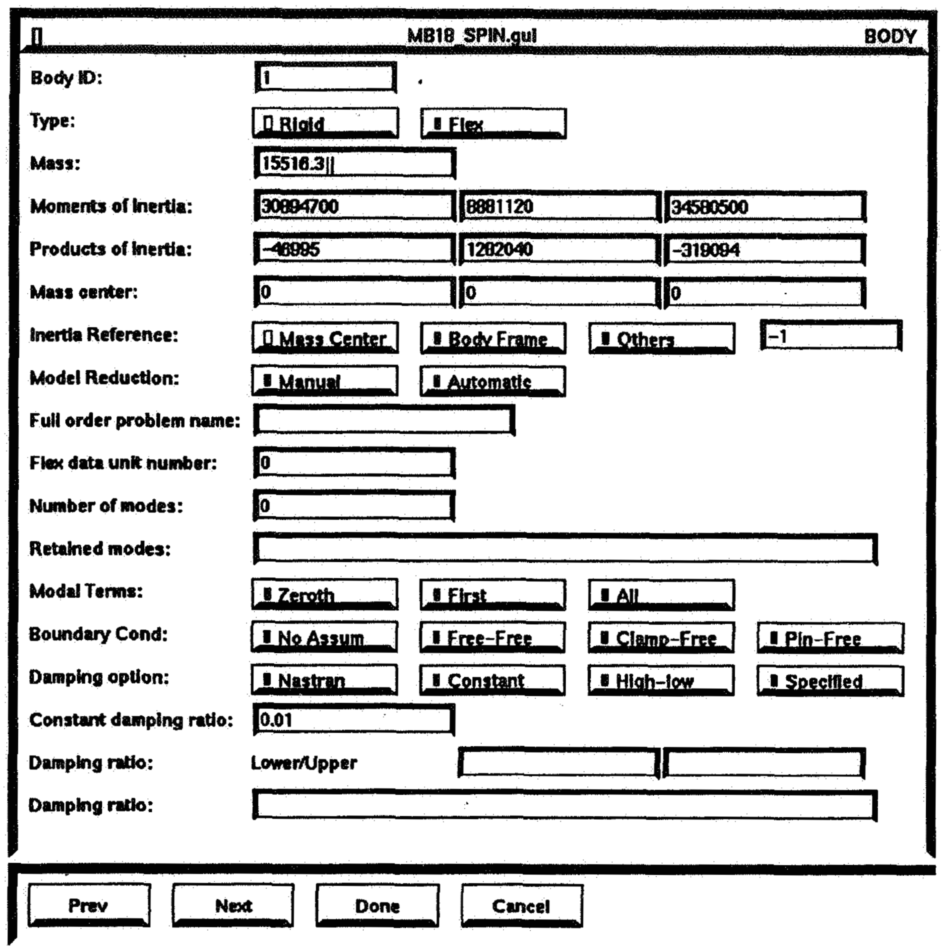

Figure 1: Data-Entry Window

geometrical representations of each body to be included in a motion picture. Finally, the procedure for providing descriptions of elastic bodies is considered.

System Topography and Mass Distribution

System topography (the manner in which bodies are connected) and mass distribution of rigid bodies are described more or less straightforwardly. One way to assess the ease or difficulty of the procedure for describing systems is to compare it to the corresponding activity performed with other multibody software, such as SD/FAST [3, p. 236]. The "free" format and use of keywords in SD/FAST system-description files circumvent many of the problems associated with the Workstation's data-entry windows. Table 1 represents a comparison of the information required to describe a rigid-body system to SD/FAST, and the Workstation; it reveals both similarities and differences.

Table 1: Information Required to Describe System

SD/FAST	Workstation	Comments
Body Name	Body Number	
Mass of Body	Mass of Body	
Three Moments of Inertia	Three Moments of Inertia	
Three Products of Inertia	Three Products of Inertia	(if necessary)
	Position vector from Q, some point on body, to $B^\star$, body mass center [default (1,1,1)].	(With SD/FAST, positions of points are measured from $B^\star$)
	"Inertia reference": Point for which inertia scalars are provided [Q (the default) or $B^\star$].	(With SD/FAST, inertia scalars for $B^\star$ must be given)
	A number to identify each node (point of interest) on each body.	Analyst can make use of predefined sensors and actuators with Workstation, but not with SD/FAST.
Inboard Joint Type (e.g. `slider`, `pin`, `u-joint`)	Inboard Joint Number. Number of • prismatic pins • revolute pins	SD/FAST has a ball joint, Workstation doesn't.
Name of Inboard Body	Numbers of *two* adjacent Bodies and *two* nodes fixed on joint.	
Position vector from $B^\star$ to J, a point fixed in adjacent bodies.	Position vector from Q to J.	
Position vector from inboard body mass center to J.	Position vector from some point on inboard body to J.	

The first item under the Workstation column that lacks a counterpart under the SD/FAST column is the position vector from Q, some point fixed in the body under consideration, to $B^\star$, the mass center of that body. With SD/FAST, the positions of all points of interest in a body are measured *from* $B^\star$; with the Workstation, the positions may be measured from any point, Q. This versatility in the Workstation can be an asset because analysts often receive data in which positions are measured from some point other than $B^\star$. However, unless Q is an interesting point in its own right, measurements from $B^\star$ are preferrable because the amount of information to be supplied is minimized. In practice, Q is often a point that does not come into play in a derivation of equations of motion. A good example of such a point is the geometric center of a space station's central truss. On the Workstation, one can provide measurements from $B^\star$, so long as the position vector from Q to $B^\star$ is made to vanish. Unfortunately, the vector from Q to $B^\star$ that is displayed when a data-entry window first appears is 1 unit in each of three directions. It is extremely unlikely that a vector of this magnitude and direction can be used in practice; one that is 0 units in each direction is more likely to result in a savings of analysts' time.

The second item without a counterpart is a pair of buttons in the data-entry window that gives analysts the option of providing inertia scalars of a body (moments and products of inertia) for either point Q or $B^\star$. *Central* inertia scalars must be used with SD/FAST; in other words, they are for $B^\star$. On the Workstation, point Q is selected when the window first appears. In practice, however, central inertia scalars are used most often, so a small amount of time can be saved if $B^\star$ is preselected by the machine.

Points that remain fixed in two adjacent bodies, points at which forces are applied, and a body's mass center, are all of special interest in connection with formulating equations of motion. The two simulation programs deal with such points in different ways. With SD/FAST, a multibody system's topography and mass distribution are described without reference to points at which forces are applied, and users do not spend time applying numeric or alphabetic labels to points that remain fixed in adjacent bodies. On the Workstation one refers to points of interest (other than mass centers of bodies) as *nodes*, and gives every node a numerical label that can serve as a shorthand for three measure numbers of a position vector. The labels provide an easy way of specifying locations of accelerometers, position sensors, reaction jets, and control-moment gyroscopes, all of which are modeled in pre-written routines available on the Workstation.

The procedure for describing joints that connect adjacent bodies is another area in which SD/FAST differs from the Workstation. Users of SD/FAST may choose a joint from a predefined set, which includes `pin`, `slider`, `u-joint`, etc. The name of an appropriate joint is registered in a system-description file. If a joint that is not a member of the set is to be used, it can be created with the joints available, and bodies without mass. On the Workstation one refers to a joint by a number; indicates whether there are 0, 1, 2, or 3 prismatic pins in the joint; and whether there are 0, 1, 2, or 3 revolute pins in the joint. The Workstation requires no more than six keystrokes: three to erase pre-existing numbers, if necessary, and three to enter the appropriate numbers. If one is constructing a system with joints unavailable in SD/FAST, the amount of labor required by SD/FAST

is obviously greater than that expended on the Workstation, but this situation is probably the exception instead of the rule. When dealing with one of SD/FAST's predefined joints, there is little difference in the level of effort put forth in this area. One joint that can be taken advantage of with version B of SD/FAST, a ball joint, is unavailable to users of the Workstation.

The different approaches to describing joints and points of interest culminates in (or follows from!) a difference in the way one indicates which bodies are connected to one another. The names of one inboard body and one inboard joint are associated with each body described to SD/FAST, with the possible exception of the base body. This information, together with two position vectors, completely describes the way in which adjacent bodies are fastened together. Instead of providing the name of an inboard body, one must, when using the Workstation, supply the numbers of *two* bodies to be connected by a joint, as well as the number of a node on *each* body. The former procedure is decidedly easier than the latter.

A description of a body's orientation is, in general, accomplished with the aid of three orthogonal, right-handed basis vectors, regarded as fixed in the body. Each basis vector associated with a body has the same direction as the corresponding vectors of every other body for the system configuration that is described to SD/FAST. All joint angles and displacements are, by definition, equal to zero in that case. Users of the Workstation do not face a similar restriction. By recording measure numbers of two orthogonal unit vectors in each of two bases fixed in adjacent bodies, an analyst can define angular displacements of a joint to be zero when both bases are not aligned. This feature can prove useful when dealing with NASTRAN information that has been produced by several people not working with a common basis.

Geometric Objects for Animations

Multibody simulations are, first and foremost, performed in order to obtain knowledge of a system's motion. Relatively large changes in position and orientation are best displayed on the Workstation in a three-dimensional animation— an extremely useful complement to a traditional two-dimensional plot, useful for depicting relatively small movements. So-called rigid body motion of a system is represented in an animation, but elastic behavior is not. The facility for animating rigid-body motion is one of the features that distinguishes the Workstation from many other simulation programs.

The apparatus for creating objects for a motion picture is not intended to take the place of a computer-aided design program. It should, however, be capable of producing simple models quickly. Plans call for the Workstation to make use of Integrated Design and Engineering Analysis Software (IDEAS) geometric models for animation, when such models are available.

Rigid geometrical objects that play the part of each body in an animation are constructed in a window that displays three orthographic projections, a perspective view, and several pull-down menus. An object appears in the orthographic projections as a wireframe. In the perspective view, an object can be shown either as a wireframe or a colored solid, and can be

seen from any angle. Each of the four views can be magnified to various levels. Models are constructed with a number of basic two-dimensional and three-dimensional objects: lines, rectangles, triangles, cubes, spheres, and cylinders. Extrusion and revolution, two laborious activities, can be used to create complex objects. The machine responds much more slowly when dealing with many objects, or objects that are complex.

New objects are placed into one of the planar views, and appear simultaneously in all views: their position, size, or orientation can be adjusted by selecting an appropriate option from one of the menus, and moving a mouse pointer to the planar view from which the changes are to be made. Adjustments can be regulated with either a mouse or arrows on the keyboard; both are usually needed because the behavior of the mouse is not easy to control, and the arrow keys act very slowly. Precise changes in an object's orientation or size are difficult to make; consequently, one is often faced with the inconvenience of repeated adjustments.

Before an object can be edited, it must first be *selected*: a procedure not easily performed on the Workstation. The most straightforward way to pick out an object would be to use a mouse to select it from one of the orthogonal projections. However, this approach has proven unsatisfactory: when several objects are close together, the wrong object is often chosen. Instead, one now works the way through three levels of menus to choose from a list of names of basic objects that have been created. The process would be speedier if the list were at a higher level. The current procedure produces lengthy delays while the computer registers a user's choice of an object. On several occasions the program has crashed while dealing with a large number of objects.

Each body is composed of one or more objects, selected and *grouped* together by a user. A body's appearance may be simple or detailed, depending on the number of objects included in the group. A number, corresponding to one of the bodies described in a data-entry window, is assigned by a user to each group. One drawback to this process is that once a group has been formally established, it can not be dissolved. Objects that are part of a group can not be edited, nor can objects be added to or removed from a body. Changing a body's appearance is cumbersome: it must be deleted entirely, and then re-created from scratch.

Up to now, the animation facility's biggest shortcoming has been the necessity for a human to perform two tasks manually. The point Q, described in the Section on system topography, had to be identified for each body. In addition, an orthogonal triad of color-coded vectors had to be correctly oriented in each body to identify the directions that were used in the descriptions in the data-entry windows. If any of this was done improperly, bodies appeared to "jump" at first, or drift apart in the course of an animation. The process involved a great deal of trial and error, and consumed much of an analyst's time. Since the computer is better suited for these jobs, its help is being enlisted in several ways.

First of all, a user assigns a scale to the grid lines in the orthogonal projections. The computer uses information from a data-entry file, together with the scale, to place graphical reference points of each body, and orient each triad of colored vectors correctly. Subsequently, an analyst adjusts the size of geometrical objects and matches up a body with

each triad. Development of automatic placement of reference points and direction vectors is continuing. Two features that might aid a user in proportioning geometric objects are a display of the mouse pointer's coordinates, and an indication of each joint's location.

Information for Elastic Bodies

In the data-entry windows discussed earlier, a body may be designated as either rigid or elastic. For each deformable body, one must provide the number of mode shapes to be used, numbers that identify the modes of interest, and a number to identify a file in which NASTRAN information for the body resides. In return, one is excused from the chore of supplying body mass, mass center position, and inertia information, all of which are contained in the NASTRAN file. In addition, the analyst indicates the nature of structural damping, and numerical values of damping ratios. The presence of *modal integrals* in equations of motion indicates coupling between rigid body motions and motions arising from a body's elasticity, as well as between motions identified with elastic modes. A user is asked to indicate whether modal integrals should contain terms in which mode shapes and slopes appear to 1st, 2nd, or 3rd order. Automatic model reduction, and boundary condition specification, which can be used to simplify equations of motion, are planned but not yet available.

Each point of interest on an elastic body must be associated with a NASTRAN grid point. In the data-entry window for each node, an analyst must record an identifying number, obtained by inspecting a table in a NASTRAN output file. This frees one from the task of furnishing a node's position, which is present in a NASTRAN file. The present Workstation user's manual does not contain a discussion of the way in which NASTRAN must be used to produce data for the Workstation. Moreover, the manual lacks information regarding the maximum number of grid points and mode shapes that can be dealt with.

Plotting and Animating Simulation Results

Time-histories of generalized coordinates, generalized speeds, and a wide variety of other simulation parameters are recorded in a file, and can be displayed as two-dimensional curves. An analyst chooses which variables to plot, and has control over details such as the number of figures on a page, and the scales of the ordinate and abscissa. A change in any *one* of these items requires that the window containing the curves be closed, and *all* of the items over which the user exercises control must be specified again. A considerable amount of time can be saved if repetitious selections are eliminated by allowing a single change to be made while the curves remain visible.

An animation is performed with a geometric model of a system and information from a numerical simulation that has been recorded in a file. Animations can be displayed in one of three orthogonal projections, the perspective view, or all four views at once. Models can be examined from any angle, and portions of any view can be magnified.

The geometric model may be displayed as a collection of wireframes or colored solids.

Animations take place considerably faster when working with the wireframes. One can control the motion picture in much the same way as one uses a video cassette recorder; playing scenes backward or forward in time, and pausing at any point. The Workstation lacks a feature that allows an animation to begin from an instant in time specified by a user. This would be useful for viewing interesting scenes without first watching preceding material.

To summarize, the animation apparatus enables one to check the reasonableness of simulation results in a way that is pleasing to the eye. It is said that a picture can be worth a thousand words: an animation can, in some cases, be worth ten or twenty plots, especially when bodies in a system undergo large changes in orientation.

Simulation Results

In order to check the simulation results given by the Workstation, and form an opinion about the ease with which the Workstation can be used, attempts have been made to re-create several simulations that have been performed by other means. Brief descriptions of those simulations, together with comparisons of the outcomes, shall be given presently. First, however, we take up two general topics related to simulation results.

Initial values of joint angles, displacements, and speeds, as well as orbital parameters, are typed into fields in the data-entry windows. Angular values must be given in units of radians. The authors, who often receive and report information in units of degrees, find this inconvenient. Plots of time histories of angular quantities are presented in radians, making it difficult to compare results to those from other simulations. It would certainly be useful to have an option for working with either unit.

Opportunities for mistakes in deriving, encoding, and numerically integrating equations of motion are legion. The use of symbol manipulation in programs like SD/FAST and AUTOLEV [4], and the Workstation's software, almost eliminate the possibility of human error. Nevertheless, it is always advisable to check the results of numerical integrations. One way of doing so is to evaluate an integral of the equations of motion, if one is available, and verify that it remains constant at every step. To this end, SD/FAST and AUTOLEV can derive and encode expressions for system central angular momentum and kinetic energy, in a Newtonian reference frame. The Workstation lacks a facility for testing results in this way, but plans call for one to be added.

Simulation of Centrifuge Operations

A Space Station Freedom centrifuge facility comprised of two rotors, each 2.5 meters in diameter, is being developed by NASA's Ames Research Center to perform experiments involving plants, rodents, and primates. The service rotor will spin up once or twice a day in order to install or remove experiments from the main rotor, which will remain spinning for about a month at a time.

Refs. [5] and [6] describe simulations carried out at NASA's Johnson Space Center to

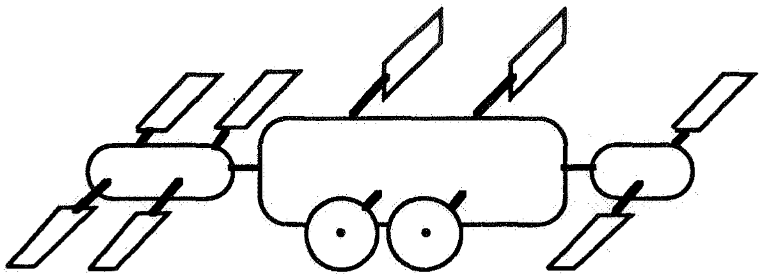

Figure 2: Space Station Topography

quantify the effects of centrifuge operations on the Station's attitude behavior and the performance of Control Moment Gyroscope (CMG) momentum management algorithms. In order to test-drive the Workstation, we attempted to duplicate a simulation very much like those mentioned in Refs. [5] and [6], which were performed with the Space Station Multi Rigid Body Simulation (SSMRBS), a collection of "user-supplied" routines written to work together with routines produced by SD/FAST [3].

The Space Station, sans the centrifuge rotors, is made up of eight bodies fastened together with simple revolute joints, as shown in Fig. 2: a core body, two truss structures immediately outboard of the core, three pairs of solar arrays attached to the outboard truss structures (two to the starboard, one to the port), and two radiator panels attached to the core body.

Each rotor is treated as a disk with uniform mass distribution, attached to the Station's core body by means of a simple revolute joint that keeps the rotor's mass center fixed in the core body. The angular displacements, speeds, and accelerations of the rotors in the core body are prescribed functions of time. As a result, several aspects of centrifuge operations are not taken into account: the change in mass distribution when experiments are moved from one rotor to another, translations of the service rotor, any mass imbalance of the rotors, and motion that results from vibration isolation mounts or centrifuge control systems. The results of interest are not likely to be altered significantly by including these details in a simulation.

The simulation, which is ten orbits in duration, is representative of much of the analysis that is done in support of the Space Station program in the area of attitude control: it involves the CMG attitude control system, controlled motion of outboard truss structures and solar arrays, and prescribed motion of appendages. The initial altitude of the Space Station is approximately 200 nautical miles, and the action of gravitational forces on each body is modeled.

The Preliminary Design Review momentum management algorithm is used to control the orientation of the Station's core body in a local-horizontal-local-vertical reference frame. The motion of each alpha and beta joint is controlled independently by means of a Proportional-Integral-Derivative (PID) feedback scheme. The Workstation subroutines associated with core body and appendage control are, as nearly as possible, identical to those

used with SSMRBS. The current Workstation handbook fails to delineate the argument list that must be present in a user-written "controller" routine. Both centrifuge rotors remain motionless for the first five orbits, after which the angular speeds of both rotors reach an absolute value of 174.3 deg/s in 130 seconds. The angular speed of the main rotor remains constant for the final five orbits, while that of the service rotor remains constant for five minutes and then returns to zero in 130 seconds.

The algorithm used to calculate gains for the PID controllers on the Workstation are slightly different from those used in SSMRBS. Furthermore, there are differences in the ephemerides used to compute the direction to the Sun. Although the simulation results from the two programs share a very strong resemblance, they are, understandably, not identical. In both cases, the system central principal axes of inertia are nearly parallel to local vertical and local horizontal, and normal to the orbit plane; the average steady-state magnitude of the sum of CMG central angular momenta is less than 3,500 ft-lb$_f$-sec, and the amplitude of the oscillations in the solar arrays' beta joints is approximately 1.5 deg. In neither case does the spin-up of the centrifuge rotors have a pronounced effect on the attitude motion of the core body.

Motions depicted in animations of the simulation results are entirely consistent with the information contained in two-dimensional plots. One interesting observation is that a viewer is presented with an optical illusion in which the main centrifuge rotor appears to spin at approximately the same speed as the outboard truss structures: about 0.064 deg/sec. Movement of the service rotor is barely noticeable. These phenomena are related to the frequency with which results are recorded. The main rotor should appear to spin faster, and more service rotor motion should be visible, if data are recorded more often. However, the animation will last longer. An animation of results from the simulation of centrifuge operations takes just over two minutes to watch when bodies are depicted as colored solids, and data is saved at intervals of 100 seconds. Means of speeding up and slowing down a motion picture would be useful, particularly when angular or linear speeds of bodies differ by, say, more than a factor of 10.

Other Rigid Body Simulations

Results from other simulations of Space Station motion, performed on the Workstation, have been compared to results obtained with Lockheed Engineering & Science Company's Station Control Simulator (SCS), which makes use of an algorithm based on Kane's method, and can simulate motion of systems with elastic bodies. The SCS algorithm is an order N variety [7], patterned after the material in Refs. [8], [9], [10], and [11].

In this Section we discuss three simulations of a rigid body spacecraft's motion, in which forces from a 55 lb$_f$ reaction control jet are applied for 200 milliseconds to the Mission-Build 5 (MB-5) configuration of Space Station Freedom, shown in Fig. 3. A fixed-step, 4th order Runge-Kutta scheme is used in all of the simulations to numerically integrate equations of motion from time $t = 0$ to $t = 20$ seconds, with a step size of 0.01 seconds. Two sensors, a rate gyroscope and an accelerometer, are placed at the center of mass of the Guidance, Navigation, and Control pallet.

Case I involves a rigid body whose mass distribution is identical to that of the MB-5 Space Station. In case II, five bodies are rigidly attached to one another: a core body, a truss structure immediately outboard of the core, two solar arrays connected to the outboard truss structure, and a radiator panel fastened to the core body. The distribution of the system's mass is the same as that of the body in case I. Clearly, a program's results for case I should be identical to those for case II. Case III differs from case II in that the five bodies are fastened together with simple revolute joints; the relative angular speed of each pair of adjacent bodies is a prescribed constant, and initial angular displacement at each joint is non-zero.

Results from simulations performed with the Workstation are in agreement with those obtained from SCS in all three cases.

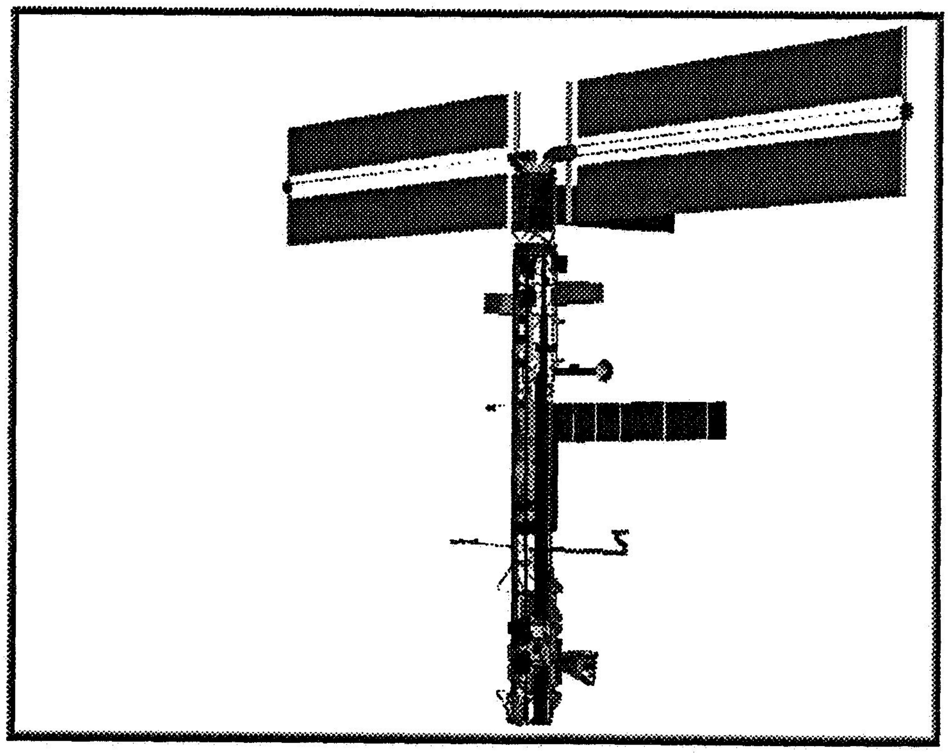

Figure 3: Space Station, Mission Build 5

Elastic Body Simulations

In this Section we compare results of simulations that are similar to those described in the preceding Section; however, the bodies are regarded as deformable.

Case IV, like case I, involves a single body. MacNeal-Schwendler Corp.'s NASTRAN has been employed to compute structure mode shapes and frequencies; 35 modes whose frequencies are below 10 Hz. have been retained. Cases V and VI are the elastic counterparts

of cases II and III, where the number of retained modes for bodies $1, \ldots, 5$ are, respectively, 12, 2, 7, 7, and 6.

The Workstation yields results that are similar to those of SCS in cases IV, V, and VI. Time histories obtained with each program are best described as oscillations superimposed on the curves obtained in the corresponding rigid-body simulations— a reassuring sign.

Computational Speed

The overriding objective in the design of Workstation hardware and software was to maximize computational speed in multibody simulations. Therefore, comparisons with other programs are once more in order. Computational tasks, with the exception of numerical integrations of equations of motion, are performed by the Workstation's Silicon Graphics Personal Iris. Integrations are handled by one or more of the four parallel processors, which are not part of the Iris' hardware.

SD/FAST uses 27.6 seconds of CPU (Central Processing Unit) time on a SUN 4/60 SPARCstation 1 to derive and encode equations of motion for the 10-body space station with centrifuge rotors. The same task is accomplished in 24.4 CPU seconds by the Personal Iris. The 10-orbit simulation is performed in 304 CPU seconds by SSMRBS (also on a SUN SPARC 1), while the Workstation takes about 100 CPU seconds. It is important to have an idea of the number of operations that can be performed over some period of time by each of the machines involved in this comparison. Our best estimate is that the SUN is capable of doing 1.4×10^6 floating point operations per second (flops). The Personal Iris accomplishes about 0.9×10^6 flops, and a Workstation processor probably carries out 10×10^6 flops for this task. The equations of motion written by our copy of SD/FAST are of order N^3, and those from the Workstation are order N, but, in this case, the Workstation makes use of only one of the four processors during the simulation. The system of interest posseses only 13 degrees of freedom, so a significant difference in the speed of the two simulations should not be expected since, as is pointed out in [8, p. 528], the order N and N^3 methods "are roughly equivalent for robots with between six to ten degrees of freedom."

Table 2 presents a comparison of computational speeds for the programs used in cases I–VI. The amount of time (in CPU seconds) required to perform the 20-second simulation is contained in the columns labeled "Sim", while the time spent prior to the simulation to process information from NASTRAN files, in cases IV–VI, is reported in the columns labeled "Flex". The machine on which the SCS resides, a Cyber 930, can perform about 1×10^6 flops. The Workstation is 2 to 3 times as fast as SCS in carrying out the rigid-body simulations, and about 4 times faster at the elastic-body tasks. It should be pointed out that the SCS does not employ symbolic manipulation to derive equations of motion; consequently, it does not enjoy the accompanying advantages in computational speed described in Ref. [3]. However, SCS is about 1.3 times faster than the Workstation at calculating modal integrals, and in case IV, 40 times as fast!

Table 2: Comparison of CPU Time, in seconds

CASE		WORKSTATION		SCS (Cyber 930)	
		Sim	Flex	Sim	Flex
I	One rigid body	87	- -	169	- -
II	One rigid body, 5 pieces	115	- -	373	- -
III	Five rigid bodies	117	- -	383	- -
IV	One elastic body	734	4792	3084	107
V	One elastic body, 5 pieces	352	314	1326	245
VI	Five elastic bodies	352	309	1331	245

Conclusions and Recommendations for Improvements

The Workstation holds a great deal of promise for expeditious simulation of motions of multibody systems: symbolic manipulation, an order N algorithm that accounts for elastic behavior, parallel processors, and a facility for animation are married together in the interest of computational speed and informative display of results. In tests performed thus far, the Workstation yields solutions that agree with those of other programs.

The advances in computational speed can be accompanied by more efficient use of analysts' time: the processes of describing a system and viewing simulation results are hampered by several aspects of Workstation software and documentation. To that end we make the following recommendations.

1. Modify certain pre-selected quantities in the data-entry windows.
2. Speed up selection of geometric objects, and make the menu more accessible.
3. Continue modifications to provide computer assistance in creating geometric models.
4. Furnish means to change motion picture speed, and to specify a time at which an animation begins.
5. Provide an option to work with angular quantities in degrees or radians.
6. Provide a facility for checking results of numerical integrations.
7. Place additional material in the User's Manual
   (a) a complete explanation of data-entry file format, and controller routine argument list.
   (b) documentation on using NASTRAN to produce data for the Workstation.
   (c) information on limitations, such as maximum number of bodies, grid points, and mode shapes.

References

[1] Kumar, M. N., and Venugopal, R., "Computational Control Workstation: Algorithms and Hardware", *Proceedings of the 5th Annual Conference on Aerospace Computational Control*, National Aeronautics and Space Administration, Jet Propulsion Laboratory, Aug. 1992.

[2] Kane, T. R., and Levinson, D. A., *Dynamics: Theory and Applications*, McGraw-Hill, New York, 1985.

[3] Rosenthal, D. E., and Sherman, M. A., "High Performance Multibody Simulations via Symbolic Equation Manipulation and Kane's Method", *The Journal of the Astronautical Sciences*, Vol. 34, No. 3, July–Sept. 1986, pp. 223–239.

[4] Schaechter, D. B., and Levinson, D. A., "Interactive Computerized Symbolic Dynamics for the Dynamicist", *The Journal of the Astronautical Sciences*, Vol. 36, No. 4, 1988, pp. 365–388.

[5] Roithmayr, C. M., "Momentum Manager Performance During Centrifuge Operations", NASA Johnson Space Center, EG2-91-051, Aug. 28, 1991.

[6] Schroeder, C. A., "Momentum Manager Performance During Centrifuge Operations", NASA Johnson Space Center, EG2-91-071, Oct. 22, 1991.

[7] Glandorf, D. R., "An Order-N Algorithm for Open Tree Structural Topologies", LESC-28665, Lockheed Engineering & Sciences Company, Houston, Texas, Jan. 1991.

[8] Rosenthal, D. E., "An Order n Formulation for Robotic Systems", *The Journal of the Astronautical Sciences*, Vol. 38, No. 4, Oct.–Dec. 1990, pp. 511–529.

[9] Rodriguez, G., "Kalman Filtering, Smoothing, and Recursive Robot-Arm Forward and Inverse Dynamics", JPL 86-48, Jet Propulsion Laboratory, Pasadena, California, Dec. 1986. Cited in Ref. [7].

[10] Rosenthal, D. E., "Order N Formulation for Equations of Motion of Multibody Systems", *Proceedings of the Workshop on Multibody Simulation*, Vol. III, National Aeronautics and Space Administration, Jet Propulsion Laboratory, April 15, 1988, pp. 1122–1150.

[11] Singh, R. P., Schubele, B., and Sunkel, J. W., "Computationally Efficient Algorithm for the Dynamics of Multi-Link Mechanisms", AIAA Paper 89-3527-CP, Guidance & Control Conference, Boston, MA, Aug. 1989. Cited in Ref. [7].